REMARKS

This application has been reviewed in light of the Office Action dated November 10, 2009. Claim 12 is pending in the application. No amendments have been made. No claims have been added or cancelled. No new matter has been added. Reconsideration of the above-identified application, in view of the following remarks, is respectfully requested.

Claim 12 of the present application stands rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 7,426,738 to Deverill et al. (hereinafter "Deverill") in view of U.S. Patent No. 6,725,243 to Snapp (hereinafter "Snapp"), U.S. Patent No. 6,738,798 to Ploetz et al. (hereinafter "Ploetz") and U.S. Patent Publication No. 2002/0124011 to Baxter et al. (hereinafter "Baxter"). This rejection is respectfully traversed.

In maintaining the rejection of Claim 12 under 35 U.S.C. § 103(a), the Examiner improperly relies on Deverill as disclosing several elements set forth in this claim. Therefore, a brief description of this reference is appropriate.

Deverill provides a system and method for measuring the latency of information flowing through a computer system (Deverill, col. 3, lines 53-55; col. 2, lines 35-38). It is explained that "latency is the total time between two measurable points and is often used to mean any delay that increases real or perceived response time" (Deverill, col. 1, lines 35-38). In order to measure the latency of information flowing through a computer system, Deverill teaches strategic insertion of software code (i.e., API code) into a computer application to mark the beginning and end of processing (Deverill, col. 7, lines 44-47). The insertion of the API code serves two purposes. First, it creates a unique identifier in order to identify the information being measured (Deverill, col. 7, lines 48-53; col. 4, lines 38-42). Deverill explains that the unique identifier can be created by aggregating the information which is inherently part of a transaction (Deverill, col. 6, line 57

– col. 7, line 18). Second, the API marks the time at which the API software code is executed and tags that time with the information being processed by the computer application (Deverill, col. 7, lines 54-57). Using the identifier and the time marks, latency can be precisely computed (Deverill, col. 8, line 20 – col. 9, line 50).

With respect to Claim 12, it is respectfully asserted that the cited references at least fail to teach or suggest *"measuring code as the code is being loaded into memory and before execution of the code"* as recited in this claim. The Examiner cites Deverill (specifically, col. 7, lines 5-17 and col. 7, line 44 – col. 8, line 31) as disclosing the italicized element set forth in Claim 12. Applicants respectfully disagree with the Examiner's interpretation of the cited passage for several reasons.

First, Deverill does not teach anything with respect to measuring "code", but rather is directed toward measuring "latency". As explained therein, "latency" is the total time between two measurable points and is often used to mean any delay that increases real or perceived response time in a computer system (Deverill, col. 1, lines 35-38). Thus, the fact that Deverill measures the delay in a computer system's response time does not in any way teach "measuring code" as recited in Claim 12.

However, assuming, arguendo, that Deverill is somehow interpreted as disclosing the measurement of code, this reference still fails to teach the above-identified element of Claim 12 since this reference does not teach or suggest that the measurements are taken "before execution of the code" as recited in this claim. The present specification explains that code is measured as the code is loaded so that the system can be properly attested and verified before the code is actually executed (see e.g., Present Specification, pg. 9, lines 2-10). To the contrary, all measurements in Deverill are taken during or after the execution of a transaction. In fact, it

would be impossible and/or illogical to interpret Deverill as measuring code before its execution since the only measurements discussed in Deverill are latency measurements (i.e., measurements of the delay associated with code being executed). In other words, given that latency measurements are essentially measurements of the delay associated with code execution, it would be impossible to take such measurements before the code had been executed. For at least this reason, Deverill cannot be interpreted as teaching or suggesting the above-identified element recited in Claim 12.

Furthermore, the Examiner's assertion that Deverill teaches *"measuring one or more parts of a server execution environment such that measurements are taken which result in a unique fingerprint for each respective selected part"* is also incorrect. Although Deverill does take latency measurements, these measurements do not "result in a unique fingerprint". Rather, it is explained in this reference that a unique identifier is created using the actual business information associated with the transaction for which latency measurements are being conducted. More specifically, Deverill explains (emphasis added):

> Unlike conventional ARM APIs, the present API does not create or pass any data from one system component to the next (e.g., a timestamp or a unique API-generated handle, correlator or other identifier) beyond that of the business information ordinarily passed in processing a transaction. **That is, the present invention recognizes that characteristic transactional information inherently associated with a given transaction, in and of itself, constitutes a readily identifiable electronic fingerprint or reference that is sufficient to enable identification and**

tracking of events processed by a computer application in executing the transaction as it flows through a computer system. For instance, characteristic business or other transactional information associated with a securities trade may include, inter alia, a Trade Identifier (or trade ID or trade reference, the identity of the party requesting the trade, the type of securities being traded, the number of securities being traded, the price of the securities, the date of the trade, whether the trade is a "buy" or a "sell", as well other trade-specific information. **Thus, the aggregation of this characteristic transactional information represents a unique identifier that itself may be directly tracked throughout processing by a computer system**, thereby eliminating the need for a new and different API-generated handle to be created, correlated and tracked at each transition from one computer system component to the next and for each computer application transaction conducted in executing the transaction. (Deverill, col. 7, line 57 – col. 8, line 18)

As explained in the passage above, the information which is ordinarily passed as part of the transaction itself (e.g., the trade ID, party identity, etc.) is used as a unique fingerprint for identifying and tracking the transaction. This is contrary to the language in Claim 12 which recites that the measurements "result in a unique fingerprint". Accordingly, for at least this reason, Deverill fails to teach or suggest "measuring one or more parts of a server execution environment such that measurements are taken which result in a unique fingerprint for each respective selected part" as recited in Claim 12.

Even further, it is respectfully asserted that the Examiner has also erred in interpreting Deverill as teaching the element of *"aggregating the unique fingerprints by an aggregation function to create an aggregated value"* as recited in Claim 12. As explained above, Deverill teaches that the information which is ordinarily passed as part of a transaction itself can be used to create a unique electronic fingerprint that can identify and track a transaction through a processing system. Moreover, as explained in the passage reproduced above, Deverill specifically teaches that a unique fingerprint can be created by aggregating such information which is inherently passed with the transaction (Deverill, col. 7, line 63 – col. 8, line 20). Although this reference teaches that a unique fingerprint can be created by aggregating such information, this reference does not teach that a plurality of unique fingerprints can be aggregated to create an aggregated value. In fact, Deverill fails to teach anything at all with respect to aggregating or combining a set of unique fingerprints. Thus, Deverill fails to teach or suggest the above-identified element recited in Claim 12.

It should be noted that the other references cited by the Examiner (i.e., Snapp, Ploetz and Baxter) fail to cure the deficiencies of Deverill with respect to the above-identified elements in Claim 12. Snapp is directed to method for accurately updating information in a database. This reference does not even remotely contemplate an attestation method which involves measuring code before execution of the code, using measurements as unique identifiers, or aggregating unique identifiers. Ploetz and Baxter likewise fail to disclose these elements. It appears these references were cited for a limited purpose. More specifically, it appears that Ploetz was cited as disclosing "if an earlier measurement exists for the code and a new measurement is different, marking the earlier measurement as changed and adding the new measurement to a list" and Baxter was cited for disclosing "wherein the measurements are stored in an order-preserving

manner in a single list". However, even if the Examiner's interpretation of these references is correct, these references still fail to teach anything with respect to measuring code before execution of the code, using measurements as unique identifiers, or aggregating unique identifiers. Therefore, since these references fail to cure the deficiencies of Deverill with respect to the above-identified elements in Claim 12, the present rejection is believed improper.

In addition to the discussion provided above, it is also pointed out that the Examiner's use and combination of the cited references is improper because the cited references are not analogous art. In *In re Kahn,* 441 F.3d 977, 987 (Fed. Cir. 2006), the Federal Circuit has explained:

> The analogous-art test requires that the Board show that a reference is either in the field of the applicant's endeavor or is reasonably pertinent to the problem with which the inventor was concerned in order to rely on that reference as a basis for rejection. *In re Oetiker,* 977 F.2d 1443, 1447 (Fed.Cir.1992). References are selected as being reasonably pertinent to the problem based on the judgment of a person having ordinary skill in the art. *Id.* ("[I]t is necessary to consider 'the reality of the circumstances,'-in other words, common sense-in deciding in which fields a person of ordinary skill would reasonably be expected to look for a solution to the problem facing the inventor." (quoting *In re Wood,* 599 F.2d 1032, 1036 (C.C.P.A.1979))).

In the present case, the invention is directed to a method of providing attestation in a server execution environment. "Attestation" is a well known term in the art for verifying the identity and integrity of a system or program running on a system. It is an important tool often used by remote clients in a client-server environment to ensure that the correct application is running on the server or that the serer has not been compromised.

While the present invention relates to an attestation method, none of the cited references involve an attestation method, or even remotely involve verifying the identity or integrity of a system or program. As explained above, Deverill, which has been cited as the primary reference, relates to a system and method for measuring the latency of information flowing through computer systems. It can hardly be argued that a latency measurement system falls within the same field as an attestation or verification system.

Furthermore, Deverill is not "reasonably pertinent" to the problem addressed by the present invention. The present specification identifies several problems associated with current attestation systems. For example, it is explained that the dynamic nature of running programs creates many problems for attestation systems (Present Specification, pg. 1, lines 12-14; pg. 2, lines 12-20), and that these problems may be further complicated when attestation systems are provided in a client-server environment (Present Specification, pg. 1, lines 14-17). However, Deverill does not teach anything with regard to these problems, or any other problems which are pertinent to the present invention. Since Deverill does not fall within the same field as the present invention and is not pertinent to any of the problems addressed by the present invention, Deverill cannot be considered analogous art and cannot be cited against the present claims.

The other cited references are related to the present invention in a manner which is even more tangential than Deverill. For example, Ploetz relates to generating reports by collecting and

summarizing data stored at a number of remote imaging devices. Snapp is directed to a method for accurately updating information in a database, and Baxter is directed to a system which uses a database interface to communicate with a controller. Like Deverill, these references do not teach anything with respect to an attestation method, nor do they relate in any way to the problems addressed by the present invention. Accordingly, it is believed that these references also pertain to non-analogous art.

In view of the foregoing, Applicants respectfully request that the rejection of Claim 12 set forth in the final Office Action of November 11, 2009 be withdrawn, that pending Claim 12 be allowed, and that the case proceed to issuance of Letters Patent in due course.

It is believed that no additional fees or charges are currently due. However, in the event that any additional fees or charges are required at this time in connection with the application, they may be charged to IBM's Deposit Account No. 50-0510.

Respectfully submitted,

Dated: 1-7-10

By: ______________________
James Bitetto
Reg. No. 40,513
Attorney for Applicant(s)

TUTUNJIAN & BITETTO, P.C.
20 Crossways Park North, Suite 210
Woodbury, NY 11797
Tel: (516) 496-3868
Fax: (516) 496-3869